Svelte Pro Techniques

Duration

2 days

Description

This Svelte Pro Techniques programming course is designed for professionals who have a basic understanding of Svelte and want to take their skills to the next level. The course covers a wide range of advanced topics including environment variables, advanced bindings, component composition, routing, actions, UI special effects, special elements, hooks, and unit testing. Participants will learn how to create complex UI effects, understand the use of special elements, and master advanced routing techniques. The course also delves into the use of hooks, server-side and client-side rendering, and unit testing. By the end of this course, participants will have a deep understanding of Svelte and be able to create complex, high-performance web applications.

Objectives

- Understand the use of environment variables in Svelte and advanced bindings including content-editable, each block bindings, media elements, and more.
- Master advanced component composition techniques such as slots, named slots, slot fallbacks, and the context API.
- Learn advanced routing techniques including optional parameters, REST parameters, param matchers, and route groups.
- Understand actions in Svelte, their uses, element-level lifecycle functions, and how to interface with 3rd party libraries.
- Learn to create UI special effects using motion, transitions, key blocks, and animations directives.
- Understand the use of special elements in Svelte such as Svelte Self, Svelte Component, Svelte Element, and more.
- Learn about hooks in Svelte, how they are used, and how to handle hook requests and errors.
- Master advanced topics such as server-side rendering, client-side rendering, pre-rendering, links, and unit testing in Svelte.

Prerequisites

All students must have Svelte, JavaScript, and HTML programming experience. Experience with CSS is helpful but not required.

Training 4 Programmers

To discuss this course and customizations:
Call: +1 434-509-5680 or Email: sales@training4programmers.com

Training Materials

All students receive comprehensive courseware covering all topics in the course. The instructor distributes courseware via GitHub. The courseware includes documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students need permission to install Node.js and Visual Studio Code on their computers. Also, students will need permission to install NPM Packages and Visual Studio Extensions. We will provide a cloud-based environment if students cannot configure a local environment.

Outline
- Introduction
- Environment Variables
- Advanced Bindings
    - Content-Editable
    - Each Block Bindings
    - Media Elements
    - Readonly Element Dimensions
    - This Bind
    - Bind to Components Props
    - Bind to Component Instances
- Advanced Component Composition
    - Slots
    - Named Slots
    - Slot Fallbacks
    - Slot Props
    - Slot Content
    - Module Context
    - Context API
- Advanced Routing
    - Optional Parameters
    - REST Parameters
    - Param Matchers
    - Route Groups
    - Organize Layouts and Routes
- Actions

- – What are Actions?
- – How are Actions used?
- – Element-Level Lifecycle Functions
- – Customizing Reusable Events
- – Interface with 3rd Party Libraries
- UI Special Effects
  - – Motion: Tweens & Spring
  - – Transitions Directive
  - – Parameters
  - – In and Out
  - – Custom CSS/JS Transitions
  - – Transition Events
  - – Key Blocks
  - – Global & Deferred Transitions
  - – Animations Directive
- Special Elements
  - – Svelte Self
  - – Svelte Component
  - – Svelte Element
  - – Svelte Window & Bindings
  - – Svelte Body
  - – Svelte Document
  - – Svelte Head
  - – Svelte Options
  - – Svelte Fragment
- Hooks
  - – What is Hook?
  - – How are Hooks used?
  - – Handle Hook
  - – Request Event
  - – Handle Fetch & Error
- Pages and Layout
  - – Server-Side Rendering
  - – Client-Side Rendering
  - – Pre-Rendering
  - – Trailing Slash

- – Universal Loading
- – Invalidation
- Links
  - – Preloading
  - – Reloading
- Unit Testing
  - – What is Unit Testing?
  - – Why Unit Test Svelte?
  - – Unit Testing Framework
  - – Testing Svelte components
  - – Testing Svelte stores
  - – Testing SvelteKit
- Conclusion